

# SPRING BEANS AUTO-WIRING

[http://www.tutorialspoint.com/spring/spring\\_beans\\_utowiring.htm](http://www.tutorialspoint.com/spring/spring_beans_utowiring.htm)

Copyright © tutorialspoint.com

You have learnt how to declare beans using the `<bean>` element and inject `<bean>` with using `<constructor-arg>` and `<property>` elements in XML configuration file.

The Spring container can **autowire** relationships between collaborating beans without using `<constructor-arg>` and `<property>` elements which helps cut down on the amount of XML configuration you write for a big Spring based application.

## Autowiring Modes:

There are following autowiring modes which can be used to instruct Spring container to use autowiring for dependency injection. You use the **autowire** attribute of the `<bean/>` element to specify autowire mode for a bean definition.

Mode	Description
no	This is default setting which means no autowiring and you should use explicit bean reference for wiring. You have nothing to do special for this wiring. This is what you already have seen in Dependency Injection chapter.
<a href="#">byName</a>	Autowiring by property name. Spring container looks at the properties of the beans on which <i>autowire</i> attribute is set to <i>byName</i> in the XML configuration file. It then tries to match and wire its properties with the beans defined by the same names in the configuration file.
<a href="#">byType</a>	Autowiring by property datatype. Spring container looks at the properties of the beans on which <i>autowire</i> attribute is set to <i>byType</i> in the XML configuration file. It then tries to match and wire a property if its <b>type</b> matches with exactly one of the beans name in configuration file. If more than one such beans exists, a fatal exception is thrown.
<a href="#">constructor</a>	Similar to <i>byType</i> , but type applies to constructor arguments. If there is not exactly one bean of the constructor argument type in the container, a fatal error is raised.
autodetect	Spring first tries to wire using autowire by <i>constructor</i> , if it does not work, Spring tries to autowire by <i>byType</i> .

You can use **byType** or **constructor** autowiring mode to wire arrays and other typed-collections.

## Limitations with autowiring:

Autowiring works best when it is used consistently across a project. If autowiring is not used in general, it might be confusing to developers to use it to wire only one or two bean definitions. Though, autowiring can significantly reduce the need to specify properties or constructor arguments but you should consider the limitations and disadvantages of autowiring before using them.

Limitations	Description
Overriding possibility	You can still specify dependencies using <code>&lt;constructor-arg&gt;</code> and <code>&lt;property&gt;</code> settings which will always override autowiring.

Primitive data types	You cannot autowire so-called simple properties such as primitives, Strings, and Classes.
Confusing nature	Autowiring is less exact than explicit wiring, so if possible prefer using explicit wiring.